REMARKS

Claims 1-21 are pending in the present application. Claims 1, 6, 10, 15, and 19 are the independent claims. In the Official Action, dated July 15, 2004, claim 10 was rejected under 35 U.S.C. § 112, ¶ 2 as lacking antecedent basis. Claims 1-21 were rejected under § 102(e) as being allegedly anticipated by U.S. Patent No. 6,145,089 (Le et al.).

Claim 6 was amended to delete the repetition of the word "caching." Claim 10 was amended to provide proper antecedent basis and was amended to bring out the inherent characteristics of one of its elements, namely that a server in a failover system is capable of processing requests for partitioned data of a respective type and that it is capable of processing requests for partitioned data other then its respective type. Claim 19 was likewise amended to bring out this inherent characteristic. All amendments in this response were made in order to more clearly communicate the subject matter of this invention, and were not made with the intention of limiting the scope of the claims.

With respect to the present Application, the format used to cite references in this response uses paragraphs (¶) and line numbers within those paragraphs that refer to the originally filed Application that can be found on the USPTO website. With respect to the cited reference (Le et al.), the standard column and line numbers are used that refer to the patent.

Rejection of Claim 10 under 35 U.S.C. § 112, ¶2

Claim 10 has been amended to provide proper antecedent basis. In the third to last element of claim 10, the second "the" and the third "the" were deleted. An "a" was inserted in front of the phrase "plurality of data servers" to properly reflect the introduction of that phrase in the claim.

Summary of the Invention

This invention relates to server failover where data is partitioned among a number of servers. For constantly changing data, the data is more typically partitioned across a number of different web servers so that each web server only handles a percentage of all the data. For

example, in a two data server scenario, the data of a first type is stored on a first server, and data of a second type is stored on a second server. Thus, the data is said to be partitioned over the first and second servers. The first server services client requests for data of the first type, whereas the second server services client requests for data of the second type. (Application, ¶ 6, lines 1-8).

In a failover scenario, when one of the these two servers is offline, the other server handles its client requests. So that when the first server is offline, the second server becomes the failover server, processing client requests for data usually cached by the first server. And vice versa. The failover server obtains the requested data from the database, temporarily caches the data, and returns the data to the requestor client. When the offline server is back online, and the failover server is notified of this, preferably the failover server then deletes the data it temporarily has cached. Needless to say, this failover procedure applies not only to a two server scenario, but to a plurality of servers so that servers can be bunched into a failover group. (Application, ¶ 8, lines 1-9, and ¶ 45 line 4-5).

In a preferred embodiment, when a server receives a client request, it first determines whether the request is for data of the type normally processed by the server. If it is, the server processes the request, returning the requested data back to the requestor client. If the data is not normally of the type processed by the server, the server determines whether the correct server to handle data of the type requested has been marked offline. If the correct server has not been marked offline, the server attempts to contact the correct server itself. If successful, the server passes the request to the correct server, which processes the request. If unsuccessful, then the server processes the request itself, querying the database for the requested data where necessary. (Application, ¶ 10, lines 1-10).

Le et al.

Le et al. deals with a method and apparatus for a server fail-over system, where a plurality of servers provide a plurality of services. The plurality of servers includes a first server for providing a first service, the system further including a client for consuming the plurality of

services, including the first service. If the first server fails to provide the first service, the first service fails over to a second server of the plurality of servers, the second server of the plurality of servers being the highest priority server for providing the first service in the event of failure of the first server. (Abstract).

Specifically, according to Figure 1B, when a failed server C is no longer accessible to the client connected to the network supported by server A, server B, and previously server C, servers A and B continue to support the services previously provided by server C. Thus, for example, while server A continues to support an intranet web server and an NFS server as it did before, after the failover it additionally supports an internet web server previously supported by server C. Selecting whether server A or server B will support the internet web server depends on the priorities of server A and server B and is determined by a nomination and an election process. (Col. 2, lines 37-63 and col. 3, lines 21-23).

For example, the role manager (RM) nominates its server, server A, for each service which server A can provide. The RM has a list of services which server A can provide. When the nomination state is completed, and all servers have converged to a single nominee server, the RM continues to proceed to the election process. The election state response depends on the results of the nomination state. There are several possible results from the nomination state: (1) time-out on the nomination process, (2) another server already has provided the service, (3) the RM is unable to provide the service, or (4) server A is selected as the server which will provide the service. This process is repeated when the next failover situation comes up. (col. 6, lines 16-63, and Figs. 6 and 9).

Rejection of Claims 1-21 under 35 U.S.C. § 102(e)

Claim 1, representative of the other independent claims, 6, 10, 15, and 19, reads as follows:

A system comprising:
a plurality of servers organized into one or more failover groups and over which *data is partitioned, each server usually processing client requests for data of a respective type* and *processing the client requests for data other than the respective type* for other of the plurality of servers within a same failover group *when the other of the plurality of servers within the same*

failover group are offline; and,

a master server managing notifications from one or more clients and from the plurality of servers as to whether servers are offline, the master server verifying whether a server is offline when so notified, and where the server has been verified as offline, so notifying the plurality of servers other than the server that has been verified as offline.

(emphasis added). Thus, each server *usually* processes *partitioned data of a respective type*, but when other servers are offline and failover occurs, servers within a failover group can *process partitioned data other than the respective type*. For constantly changing data, the data is partitioned across a number of different web servers. Thus, each server handles only a percentage of all the data, and preferably each server only caches its respective data. The advantage provided by the present invention over all prior art is, that *in a partitioned data setting*, servers that typically process such *partitioned data* of some *respective type*, also process *partitioned data other* than the *respective type* when failover occurs.

Le et al. discloses a failover system that includes a plurality of servers for providing a plurality of services. But unlike claim 1, Le et al. merely discloses the continuance of supporting services of one server by another server, such as continuing to support the original intranet web server and the NFS server in addition to the new internet web server, as can be seen in Fig. 1B. This disclosure fails to teach "*data [that] is **partitioned**, [so that] each server [is] usually processing client requests for **data of a respective type*** and *processing the client requests for data **other than the respective type*** ... when the other of the plurality of servers within the same failover group are offline." Claim 1 (emphasis added).

Accordingly, it is respectfully submitted that claim 1, representative of the other independent claims, 6, 10, 15, and 19, is patentable over Le et al. Since dependent claims 2-5, 7-9, 11-14, 16-18, and 20-21 depend either directly or indirectly from independent claims 1, 6, 10, 15, and 19, respectively, they are believed allowable for the same reasons. Withdrawal of the rejection under § 102(e) is therefore earnestly solicited.

CONCLUSION

Applicants believe that the present Amendment is responsive to each of the points raised by the Examiner in the Official action, and submits that Claims 1-21 of the application are in condition for allowance. Favorable consideration and passage to issue of the application at the Examiner's earliest convenience is earnestly solicited.

Date: October 14, 2004

Grzegorz S. Plichta
Registration No. 55,541

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439